IN THE CLAIMS

1. (**Currently Amended**) A method of recognizing words, comprising:

accepting a stroke as an input on a virtual keyboard;

defining word patterns of a plurality of known words by a plurality of paths, wherein each path connects elements in [[a]] the known word on [[a]] the virtual keyboard, wherein the virtual keyboard ~~contains a set of~~ comprises virtual keys, each virtual key representing a letter ~~characters forming elements~~ in [[the]] a word without a temporary target ~~letters~~ letter being placed adjacent to a ~~current stroke~~ location of a stroke;

~~accepting a stroke as an input on the virtual keyboard layout;~~

processing the stroke using a combination of a plurality of channels, each channel selectively measuring a different aspect of ~~the stroke's~~ a similarity of the stroke to [[the]] a plurality of [[the paths]] possible paths on the virtual keyboard; ~~and~~

converting each different aspect of the stroke's similarity to probability estimates;

a shape channel of the plurality of channels measuring a shape aspect of the stroke, and outputting a probability estimate;

a location channel of the plurality of channels measuring location aspect of the stroke, and outputting a probability estimate, wherein the location channel measures the location aspect of the stroke concurrently with the shape channel measuring the shape aspect of the stroke;

mathematically integrating, using Bayes' theorem, the probability estimates of the plurality of channels to produce integrated probability estimates of candidate words corresponding to the stroke; and

based on the integrated probability estimates of the candidate words, recognizing the stroke as a known word.

2. (**Currently Amended**) The method of claim 1, wherein [[one]] the shape channel ~~of the plurality of channels comprises~~ outputs normalized shape information independent of location and scale.

3. (**Canceled**)

4. (**Currently Amended**) The method of claim 1, wherein the plurality of channels ~~comprise~~ comprises a tunnel model channel, wherein the tunnel of a word pattern comprises a predetermined width on either side of a set of the virtual keys representing a set of letters of a word on the virtual keyboard, and wherein the tunnel model channel is applied to the stroke before any other channel is applied to the stroke.

5. (**Currently Amended**) The method of claim 1, wherein the plurality of channels ~~comprise~~ comprises a language context channel that stores recognized known words, and wherein the language context channel provides clues for recognizing a word based on a stored previously recognized known word.

6. (**Currently Amended**) The method of claim 2, wherein recognizing a word pattern using the normalized shape information comprises at least one of: template matching and feature extraction.

7. (**Canceled**)

8. (**Currently Amended**) The method of claim [[3]] 1, wherein the location channel ~~recognizing~~ recognizes a word pattern ~~using path location information comprises using location matching, wherein~~ ~~location matching comprises~~ by sampling a plurality of points on the stroke, including at least one sampling point between a beginning and an end of the stroke, each sampling point having a weight, and by applying weights to ~~weighting~~ sampling points of ~~location from beginning to end~~ the stroke, wherein each sampling point has a different weight, and wherein a sampling point at the beginning of the stroke has a greatest weight and a sampling point at the end of the stroke has a least weight.

9. (Previously Canceled)

10. (**Canceled**)

11. (Previously Presented) The method of claim 4, wherein recognizing a word pattern using the tunnel model channel comprises traversing keys passed by the word pattern and identifying potential word candidates by partial string matching.

12. (**Currently Amended**) The method of claim 4, wherein recognizing a word pattern using the tunnel model channel comprises transforming a tunnel and ~~a gesture~~ the stroke passing the tunnel.

13. (**Canceled**)

14. (**Currently Amended**) The method of claim [[13]] 2, further comprising: ~~displaying the~~

generating an intermediate shape that represents a difference between the ~~user's gesture trace~~ stroke and [[the]] an ideal template of the word pattern by morphing the ~~user's gesture trace to the~~ stroke with an ideal template; and

displaying the intermediate shape.

15. (Previously Canceled)

16. (Previously Canceled)

17. (Previously Presented) The method of claim 1, further comprising analyzing the stroke to differentiate between a tapping and a shorthand gesture input; and inputting at least one letter of a word by tapping the letter.

18. (**Currently Amended**) The method of claim [[13]] 2, further comprising:

comparing a normalized word pattern and a normalized ~~gesture trace~~ stroke; ~~and~~

sampling the normalized word pattern ~~and gesture trace~~ to a fixed number of a plurality of points;

sampling the stroke to a same fixed number of a plurality of points;

and measuring the plurality of points relative to each other.

19. (**Currently Amended**) The method of claim [[13]] 14, further comprising:

comparing a feature vector of the ~~gesture trace and the~~ stroke with a feature vector of [[a]] the ideal template of the word pattern;

computing a similarity score from said comparing; and

obtaining a distance measurement between the stroke and the ideal template of the word pattern from said similarity score.

20. (Previously Canceled)

21. (**Currently Amended**) A shorthand symbol system for recognizing words, comprising:

a graphical keyboard ~~layer~~ layout for accepting a stroke as an input ~~trace~~, wherein the keyboard ~~layer~~ layout contains a set of characters forming elements in the word without a temporary target ~~letters~~ element being placed adjacent to a current stroke location;

a storage for storing word patterns of a plurality of paths, wherein each path connects a set of letters received from the graphical keyboard ~~layer~~ layout;

a pattern recognition engine that recognizes a word pattern by processing the stroke using a combination of a plurality of channels, each channel selectively processing, in parallel, a different aspect of the ~~input trace~~ stroke in relation to the plurality of the paths on the graphical keyboard ~~layer~~ layout and producing an output representing a probability estimate for a candidate word, one channel of the plurality of channels processing a location-based similarity probability estimate, another channel of the plurality of channels processing a shape-based similarity probability estimate, still another channel of the plurality of channels processing a path-based similarity probability estimate, and yet another channel of the plurality of channels processing a language context-based similarity probability estimate; and

a computer for ~~mathematically integrating outputs of the plurality of channels to produce an integrated~~ producing a probability estimate of a candidate word, wherein the computer produces the probability estimate of the candidate word by first serially applying the output of each channel of the plurality of channels alone and separately, and if a recognized word cannot be identified from the output of any one channel, then the computer mathematically integrates outputs of at least two channels of the plurality of channels to produce an integrated probability estimate of the candidate word.

22. (**Currently Amended**) The method of claim 21, wherein [[one]] the channel ~~of the plurality of channels comprises~~ that processes a shape-based similarity probability estimate outputs normalized shape information independent of location and scale.

23. (**Currently Amended**) The method of claim 21, wherein the one channel of the plurality of channels comprises ~~path~~ location information regarding sampling points of the stroke, wherein each sampling point ~~having~~ has a different weight.

24. (**Canceled**)

25. (**Canceled**)

26. (**Canceled**)

27. (Previously Presented) The system of claim 21, wherein the word patterns comprise letters from an alphabet.

28. (Previously Presented) The system of claim 21, wherein the word patterns comprise letters from Chinese pinyin characters.

29. (Previously Canceled)

30. (**Currently Amended**) The system of claim 21, wherein the word patterns are based on a lexicon, and wherein the lexicon comprises a very large collection of words used in a natural language, and wherein words in the lexicon are rank ordered by usage frequency, and more frequent words are given higher *a priori* probability.

31. (**Canceled**)

32. (Previously Presented) The system of claim 21, wherein the word patterns are based on a lexicon, wherein the lexicon is customized from an individual user's previous documents for a specific application, and wherein part of the customized lexicon is based on a computer programming language.

33. (Previously Canceled)

34. (Previously Canceled).

35. (Previously Presented) The system of claim 21, wherein the word patterns are based on a lexicon, and wherein the lexicon is customized for a specific domain.

36. (Previously Canceled)

37. (**Currently Amended**) A method of recognizing words, the method comprising:

defining word patterns of a plurality of known words by a plurality of paths, wherein each path connects elements in a word on a virtual keyboard, wherein the virtual keyboard ~~contains a set of~~ comprises virtual keys, each virtual key representing a character, each character ~~characters~~ forming ~~elements~~ an element in [[the]] a word without a temporary target ~~letters~~ element being placed adjacent to a current stroke location;

accepting a stroke that represents one whole word as an input on the virtual keyboard ~~layout~~;

recognizing a word pattern by processing the stroke using at least one location channel that selectively process different aspects of the stroke in relation to the plurality of the paths on the virtual keyboard, the at least one location channel processing a location-based similarity probability estimate;

determining a time spent inputting the stroke; and

modifying the location-based similarity probability estimate according to a path of the stroke on the virtual keyboard and the time spent inputting the stroke, to produce an output of the at least one location channel.

38. (**Currently Amended**) The method of claim 37, ~~further comprising~~ including recognizing a word pattern by processing the stroke using the location channel and any one or more of: a channel for shape information, a channel for a tunnel model, and a channel for a language context.

39. (Previously Canceled)

40. **(Canceled)**

41. **(Canceled)**

42. **(Canceled)**

43. (Previously Presented) The method of claim 1, including ranking the candidate words in order of probability.

44. (Previously Presented) The method of claim 1, including:

determining a time spent inputting the stroke; and

modifying at least one probability estimate according to a path of the stroke on the virtual keyboard and the time spent inputting the stroke, to produce an output of at least one channel of the plurality of channels.

45. (**Currently Amended**) A method of recognizing words, comprising:

defining word patterns of a plurality of known words by a plurality of paths, wherein each path connects elements in [[a]] the known word on a virtual keyboard, wherein the virtual keyboard ~~contains a set of~~ comprises virtual keys, each virtual key representing a character, each character ~~characters~~ forming ~~elements~~ an element in [[the]] a word without a temporary target ~~letters~~ element being placed adjacent to a current stroke location;

accepting a stroke as an input on the virtual keyboard ~~layout~~, wherein the stroke represents exactly one word;

processing the stroke using a combination of a plurality of channels, each channel selectively measuring a different aspect of the stroke's similarity to the plurality of the word paths on the virtual keyboard;

using path location regarding sampling points of the stroke as one channel of the plurality of channels;

using normalized path shape independent of location and scale as another channel of the plurality of channels;

converting each different aspect of ~~the stroke's~~ a similarity of the stroke to a known word to probability estimates;

measuring time spent on inputting the stroke; and

mathematically integrating the probability estimates of the plurality of channels to produce integrated probability estimates of candidate words,

wherein the time information is used to adjust a relative weight between the path location channel and the normalized path shape channel ~~in the mathematical integration of~~ when mathematically integrating the probability estimates of the ~~two~~ plurality of channels.

46. (**Currently Amended**) The method of claim 45, wherein yet another channel of the plurality of channels comprises a tunnel model channel, and wherein a tunnel of a word pattern comprises a predetermined width on either side of a set of the virtual keys representing a set of ~~letters~~ elements of [[the]] a word on [[a]] the virtual keyboard.

47. (Previously Presented) The method of claim 46, wherein recognizing a word pattern using the tunnel model channel comprises traversing keys passed by the word pattern and identifying potential word candidates by partial string matching.

48. (**Currently Amended**) The method of claim 47, wherein recognizing a word pattern using the tunnel model channel comprises transforming a tunnel and ~~a gesture~~ the stroke passing the tunnel.

49. (**Currently Amended**) A method of recognizing words, comprising:

defining word patterns of a plurality of known words by a plurality of paths, wherein each path connects elements in [[a]] the known word on a virtual keyboard, wherein the virtual keyboard ~~contains a set of~~ comprises virtual keys, each virtual key representing a character, each character ~~characters~~ forming ~~elements~~ an element in [[the]] a word ~~without temporary target letters being placed adjacent to a current stroke location~~;

accepting a stroke as [[an input]] a candidate word inputted on the virtual keyboard ~~layout~~;

recognizing a word pattern by processing the stroke using a combination of a plurality of channels, each channel selectively processing a different aspect of the stroke ~~in relation to the plurality of the paths on the virtual keyboard~~, one channel of the plurality of channels ~~processing~~ determining a weighted location-based similarity probability estimate from a location-based similarity probability estimate;

determining a time spent inputting the stroke;

modifying the weighted location-based similarity probability estimate according to a path of the stroke on the virtual keyboard and the time spent inputting the stroke, to produce an output of the one channel, wherein modifying further comprises:

calculating a total normative time of inputting the stroke for each word *i*, as follows:

$$t_n(i) = na + b\sum_{k=1}^{n-1}\log_2\left(\frac{D_{k,k+1}}{W}+1\right)$$

where $D_{k,k+1}$ is a distance between the kth and the (k+1)th letters of word *i* on the keyboard; *W* is a key width, n is a number of letters in the word; and a and b are two constants in Fitts' law,

calculating a total normative time of inputting the stroke for all words of a gesture production, as follows:

$$t_a = \Sigma\, t_n(i)$$

and if $t_a \leq t_n(i)$, then a ratio $t_n(i)/t_a$ is used to adjust distribution of the probability estimates so as to lower the weight of the location channel; and

mathematically integrating outputs of the plurality of channels to produce an integrated probability estimate of the candidate ~~words~~ word.

50. (Previously Presented) The method of claim 49, including the step of ranking the candidate words in order of probability.

51. (Previously Presented) The method of claim 49, wherein another channel of the plurality of channels comprise a tunnel model channel.

52. (Previously Presented) The method of claim 51, wherein still another channel of the plurality of channels comprise a language context channel.

53. (Previously Presented) The method of claim 52, wherein yet another channel of the plurality of channels comprises shape information.

54. (Previously Presented) The method of claim 53, wherein recognizing a word pattern using the shape information comprises template matching.

55. (Previously Presented) The method of claim 54, wherein recognizing a word pattern using the shape information comprises feature extraction.

56. (**New**) The method of claim 1, including, prior to mathematically integrating, the steps of:

setting a threshold for probability estimates; and

pruning words whose probability estimates are lower than the threshold.

57. (**New**) The method of claim 1, wherein the shape channel outputs for each word a probability estimate $x_S(i)$, and wherein the location channel outputs for each word a probability estimate $x_L(i)$, and including, for each word, the step of:

mathematically integrating the probability estimates of the plurality of channels first by producing for each channel a score

$$y(i) \in [0,1]: y(i) = e^{-x(i)/\theta},$$

where y is a variable between 0 and 1, and where θ is a weighting coefficient, and second by adding $y_S(i)$; and $y_L(i)$ such that the sum is an integrated probability estimate y(i) of a candidate word.

58. (**New**) The method of claim 57, including pruning all scores $y(i) < 0.04$ prior to adding $y_S(i)$ and $y_L(i)$.

59. (**New**) The method of claim 57, including the steps of:

setting a threshold for probability estimates;

pruning words whose probability estimates are lower than the threshold;

calculating a probability *p* of a word *i* based on values of x_S and x_L provided by the shape channel and the location channel, respectively, for those candidate words for the stroke that have not been pruned ($i \in W$) as follows,

$$p(i) = \frac{y(i)}{\sum_{i \in W} y(i)}$$ and

after calculating p_S from the shape channel and p_L from the location channel, integrating candidate words from the shape channel and the location channel as follows,

$$p(i) = \frac{p_s(i)\, p_l(i)}{\sum_{j \in W_s \cap W_l} p_s(j)\, p_l(j)}$$

where p_S is the probability estimate from the shape channel, p_L is the probability estimate from the location channel, W_S is a set of candidate words outputted from the shape channel, and W_L is a set of candidate words outputted from the shape channel.

60. (**New**) The method of claim 1, wherein the location channel places varied weights on different sampling points of the stroke as follows:

$$x_L(i)=\sum_{k=0}^{N}\alpha(k)d_2(k)$$

where $\alpha(k)$ is a relative weight placed on a k^{th} sampling point of the stroke (from $k = 0$ to $k = N$), where d_2 is a distance between a point of the stroke and a corresponding point of a template, and where $x_L(i)$ is a probability estimate that the stroke is word *i* intended by a user who makes the stroke.

61. (**New**) The method of claim 60, wherein $\alpha(k)$ is calculated as follows:

$$\alpha(k)=\frac{N-k(1-\beta)}{(1+N)N-(1-\beta)\sum_{k=1}^{N}k}$$

where $\beta = \alpha(0) / \alpha(N)$, and where $\alpha(0) + \alpha(1) + \alpha(2) + \ldots + \alpha(k) + \ldots + \alpha(N) = 1$.

62. (**New**) The method of claim 4, wherein the tunnel model channel includes two stages, a first stage generates a small candidate list efficiently by partial string matching, and a second stage verifies the word candidate by checking whether all of the points in a stroke falls into a virtual tunnel.

63. **(New)** The method of claim 62, wherein the first stage of the tunnel model channel includes

a first step that sequentially compares all the characters in the stroke with all the characters in templates in a lexicon to match all the characters in the known word,

a second step in which captured points from the stroke are translated to raw trace characters one by one, according to the virtual button at which the point is located,

a third step in which adjacent duplicate characters as well as none alphabetic characters in the raw trace characters are removed to generate a trace string, and

a fourth step in which each of the words in the lexicon is matched with the trace string to verify whether all the characters of a word appears in the trace string sequentially, and if so, the tunnel model channel saves that word in a word candidate list.

64. **(New)** The method of claim 63, wherein the second stage of the tunnel model channel includes

a first step that constructs a corresponding tunnel model for each word in the word candidate list,

a second step in which all the words in the word candidate list are tested to determine whether the stroke exceed the tunnel boundaries of a corresponding known word, and

a third step in which each point in the stroke is checked to determine whether it stays within a word tunnel, and if so, that word is selected as an output of the tunnel channel model.

65. **(New)** The method of claim 49, wherein the step of recognizing includes determining a weighted location-based similarity probability estimate y(i) from a location-based similarity probability estimate x(i), as follows

$$y(i) \in [0,1]: y(i) = e^{-x(i)/\theta},$$

where y is a variable between 0 and 1, and where θ is a weighting coefficient.